EXAMPLE - TAYLOR SERIES METHOD

Consider solving

$$y' = y \cos x, \quad y(0) = 1$$

Imagine writing a Taylor series for the solution $Y(x)$, say initially about $x = 0$. Then

$$Y(h) = Y(0) + hY'(0) + \frac{h^2}{2}Y''(0) + \frac{h^3}{6}Y'''(0) + \cdots$$

We can calculate $Y'(0) = Y(0)\cos(0) = 1$. How do we calculate $Y''(0)$ and higher order derivatives?

$$Y'(x) = Y(x)\cos(x)$$
$$Y''(x) = -Y(x)\sin(x) + Y'(x)\cos(x)$$
$$Y'''(x) = -Y(x)\cos(x) - 2Y'(x)\sin(x) + Y''(x)\cos x$$

Then

$$Y''(0) = -Y(0)\sin(0) + Y'(0)\cos(0) = 1$$
$$Y'''(0) = -Y(0)\cos(0) - 2Y'(0)\sin(0) + Y''(0)\cos 0 = 0$$

Thus

$$\begin{aligned} Y(h) &= Y(0) + hY'(0) + \tfrac{h^2}{2}Y''(0) \\ &\quad + \tfrac{h^3}{6}Y'''(0) + \cdots \\ &= 1 + h + \tfrac{h^2}{2} + \cdots \end{aligned}$$

We can generate as many terms as desired, obtaining added accuracy as we do so. In this particular case, the true solution is $Y(x) = \exp(\sin x)$. Thus

$$Y(h) = 1 + h + \frac{h^2}{2} - \frac{h^4}{8} + \cdots$$

We can truncate the series after a particular order. Then continue with the same process to generate approximations to $Y(2h), Y(3h), \ldots$ Letting $x_n = nh$, and using the order 2 Taylor approximation, we have

$$Y(x_{n+1}) = Y(x_n) + hY'(x_n) + \frac{h^2}{2}Y''(x_n) + \frac{h^3}{6}Y'''(\xi_n)$$

with $x_n \leq \xi_n \leq x_{n+1}$. Drop the truncation error term, and then define

$$y_{n+1} = y_n + hy'_n + \frac{h^2}{2}y''_n, \quad n \geq 0$$

with

$$y'_n = y_n \cos(x_n)$$

$$y''_n = -y_n \sin(x_n) + y'_n \cos(x_n)$$

We give a numerical example of computing the numerical solution with Taylor series methods of orders 2, 3, and 4.

A 4^{th}-ORDER EXAMPLE

Consider solving

$$y' = -y, \quad y(0) = 1$$

whose true solution is $Y(x) = e^{-x}$. Differentiating the equation

$$Y'(x) = -Y(x)$$

we obtain

$$Y'' = -Y' = Y$$
$$Y''' = Y' = -Y, \quad Y^{(4)} = Y$$

Then expanding $Y(x_n + h)$ in a Taylor series,

$$Y(x_{n+1}) = Y_n + hY'_n + \frac{h^2}{2}Y''_n + \frac{h^3}{6}Y'''_n + \frac{h^4}{24}Y_n^{(4)} + \frac{h^5}{120}Y^{(4)}(\xi_n)$$

Dropping the truncation error, we have the numerical method

$$\begin{aligned} y_{n+1} &= y_n + hy'_n + \frac{h^2}{2}y''_n + \frac{h^3}{6}y'''_n + \frac{h^4}{24}y_n^{(4)} \\ &= \left(1 - h + \frac{h^2}{2} - \frac{h^3}{6} + \frac{h^4}{24}\right) y_n \end{aligned}$$

with $y_0 = 1$. By induction,

$$y_n = \left(1 - h + \frac{h^2}{2} - \frac{h^3}{6} + \frac{h^4}{24}\right)^n, \quad n \geq 0$$

Recall that

$$e^{-h} = 1 - h + \frac{h^2}{2} - \frac{h^3}{6} + \frac{h^4}{24} - \frac{h^5}{120}e^{-\xi}$$

with $0 < \xi < h$. Then

$$\begin{aligned} y_n &= \left(e^{-h} + \frac{h^5}{120}e^{-\xi}\right)^n \\ &= e^{-nh}\left(1 + \frac{h^5}{120}e^{h-\xi}\right)^n \\ &\doteq e^{-x_n}\left(1 + \frac{x_n h^4}{120}e^{h-\xi}\right) \end{aligned}$$

Thus

$$Y(x_n) - y_n = x_n e^{-x_n} \cdot O(h^4)$$

ERROR

Returning to the preceding order 2 approximation, we have

$$Y(x_{n+1}) = Y(x_n) + hY'(x_n) + \frac{h^2}{2}Y''(x_n) + \frac{h^3}{6}Y'''(\xi_n)$$
$$y_{n+1} = y_n + hy'_n + \frac{h^2}{2}y''_n, \quad n \geq 0$$

Subtracting,

$$e_{n+1} = e_n + he'_n + \frac{h^2}{2}e''_n + \frac{h^3}{6}Y'''(\xi_n)$$

$$\begin{aligned} e'_n &= Y_n \cos(x_n) - y_n \cos(x_n) \\ &= e_n \cos(x_n) \\ |e'_n| &\leq |e_n| \end{aligned}$$

Similarly,

$$\begin{aligned} e''_n &= -e_n \sin(x_n) + e'_n \cos(x_n) \\ |e''_n| &\leq |e_n| + |e'_n| \\ &\leq 2\,|e_n| \end{aligned}$$

Returning to the error equation

$$e_{n+1} = e_n + he'_n + \frac{h^2}{2}e''_n + \frac{h^3}{6}Y'''(\xi_n)$$

we have

$$\begin{aligned} |e_{n+1}| &\leq |e_n| + h\,|e'_n| + \tfrac{h^2}{2}\,|e''_n| + \tfrac{h^3}{6}\,|Y'''(\xi_n)| \\ &\leq \left[1 + h + h^2\right]|e_n| + \tfrac{h^3}{6}\,\|Y'''\|_\infty \end{aligned}$$

We can now imitate the proof of convergence for Euler's method, obtaining

$$|e_n| \leq e^{2x_n}\,|e_0| + h^2\frac{e^{2x_n} - 1}{12}\left\|Y'''\right\|_\infty$$

provided $h \leq 1$. Thus

$$|Y(x_n) - y_h(x_n)| \leq O(h^2)$$

By similar means, we can show that for the Taylor series method of order r, the method will converge with

$$|Y(x_n) - y_h(x_n)| \leq O(h^r)$$

We can introduce the Taylor series method for the general problem

$$y' = f(x, y), \quad y(x_0) = Y_0$$

Simply imitiate what was done above for the particular problem $y' = y \cos x$.
In general,

$$\begin{aligned} Y'(x) &= f\left(x, Y(x)\right) \\ Y''(x) &= f_x\left(x, Y(x)\right) + f_y\left(x, Y(x)\right) Y'(x) \\ &= f_x\left(x, Y(x)\right) + f_y\left(x, Y(x)\right) f\left(x, Y(x)\right) \\ Y'''(x) &= f_{xx} + 2 f_{xy} f + f_{yy} f^2 + f_y f_x + f_y^2 f \end{aligned}$$

and we can continue on this manner. Thus we can calculate derivatives of any order for $Y(x)$; and then we can define Taylor series of any desired order.
This used to be considered much too arduous a task for practical problems, because everything had to be done by hand. But with symbolic programs such as *Mathematica* and *Maple*, Taylor series can be considered a serious framework for numerical methods. Programs that implement this in an automatic way, with varying order and stepsize, are available.

RUNGE-KUTTA METHODS

Nonetheless, most researchers still consider Taylor series methods to be too expensive for most practical problems (a point contested by others). This leads us to look for other one-step methods which imitate the Taylor series methods, without the necessity to calculate the higher order derivatives. These are called *Runge-Kutta methods*. There are a number of ways in which one can approach Runge-Kutta methods, and I will follow a fairly classical approach. Later I may introduce some other approaches to the development of such methods.

We begin by considering *explicit Runge-Kutta methods of order 2*. We want to write

$$y_{n+1} = y_n + hF(x_n, y_n, h; f)$$

with $F(x_n, y_n, h; f)$ some carefully chosen approximation to $f(x, y)$ on the interval $[x_n, x_{n+1}]$. In particular, write

$$\begin{aligned} &F(x, y, h; f) \\ &\quad = \gamma_1 f(x, y) + \gamma_2 f(x + \alpha h, y + \beta h f(x, y)) \end{aligned}$$

This is some kind of "average" derivative. Intuitively, we should restrict α so that $0 \leq \alpha \leq 1$. Otherwise, how should the constants $\gamma_1, \gamma_2, \alpha, \beta$ be chosen?

We attempt to make the leading terms in the Taylor series of $Y(x + h)$ look like the corresponding terms in a Taylor series expansion of

$$Y(x) + hF(x, Y(x), h; f)$$

More precisely, introduce the truncation error

$$T_n(Y) = Y(x + h) - [Y(x) + hF(x, Y(x), h; f)]$$

Expand $Y(x+h)$, obtaining

$$\begin{aligned} Y(x+h) &= Y(x) + hY'(x) + \tfrac{h^2}{2}Y''(x) + \cdots \\ &= Y(x) + hf + \tfrac{h^2}{2}\left[f_x + f_y f\right] \\ &\quad + \tfrac{h^3}{6}\left[f_{xx} + 2f_{xy}f + f_{yy}f^2 + f_y f_x + f_y^2 f\right] \end{aligned}$$

in which all functions f, f_x, etc. are evaluated at $(x, Y(x))$.

Next expand $f(x+\alpha h, Y(x) + \beta h f(x, Y(x)))$ in a series in powers of h. This requires the multivariable Taylor series:

$$\begin{aligned} &f(a+\delta, b+\eta) \\ &\quad = \sum_{j=0}^{r} \tfrac{1}{j!}\left(\delta\tfrac{\partial}{\partial x} + \eta\tfrac{\partial}{\partial z}\right)^j f(x,z)\Big|_{\substack{x=a \\ z=b}} \\ &\quad + \tfrac{1}{(r+1)!}\left(\delta\tfrac{\partial}{\partial x} + \eta\tfrac{\partial}{\partial z}\right)^{r+1} f(x,z)\Big|_{\substack{x=a+\theta\delta \\ z=b+\theta\eta}} \end{aligned}$$

in which $0 < \theta < 1$. Use this with $a = x$, $b = Y(x)$, $\delta = \alpha h$, and $\eta = \beta h f(x, Y(x))$.

We write out the first few terms:

$$
\begin{aligned}
&f(x+\alpha h, Y(x)+\beta h f(x, Y(x))) \\
&= f(x, Y(x)) + \alpha h f_x + \beta h f_y f \\
&+\tfrac{1}{2}\left[(\alpha h)^2 f_{xx} + 2(\alpha h)(\beta h f) f_{xy} + (\beta h f)^2 f_{yy}\right] + \cdots
\end{aligned}
$$

in which all terms f, f_x, f_y, etc. are evaluated at $(x, Y(x))$. Introduce this into the formula

$$
\begin{aligned}
&F(x, Y(x), h; f) \\
&\quad = \gamma_1 f(x, Y(x)) + \gamma_2 f(x+\alpha h, Y(x)+\beta h f(x, Y(x)))
\end{aligned}
$$

and collect together common powers of h. Then substitute this into

$$
T_n(Y) = Y(x+h) - [Y(x) + hF(x, Y(x), h; f)]
$$

and also substitute the earlier expansion of $Y(x+h)$.

This leads to the formula

$$T_n(y) = c_1 h + c_2 h^2 + c_3 h^3 + \cdots$$

$$c_1 = [1 - \gamma_1 - \gamma_2]\, f(x, Y(x))$$

$$c_2 = \left(\frac{1}{2} - \gamma_2 \alpha\right) f_x + \left(\frac{1}{2} - \gamma_2 \beta\right) f_y f$$

$$\begin{aligned} c_3 \;\; = \;\; & \left(\tfrac{1}{6} - \tfrac{1}{2}\gamma_2 \alpha^2\right) f_{xx} + \left(\tfrac{1}{3} - \gamma_2 \alpha\beta\right) f_{xy} f \\ & + \left(\tfrac{1}{6} - \tfrac{1}{2}\gamma_2 \beta^2\right) f_{yy} f^2 + \tfrac{1}{6} f_y f_x + \tfrac{1}{6} f_y^2 f \end{aligned}$$

We try to set to zero as many as possible of the coefficients c_1, c_2, c_3,... Note that c_3 cannot be made to equal zero in general, because of the final terms

$$\frac{1}{6} f_y f_x + \frac{1}{6} f_y^2 f$$

which are independent of the choice of the coefficients $\gamma_1, \gamma_2, \alpha, \beta$.

We can make both c_1 and c_2 zero by requiring

$$\begin{aligned} 1-\gamma_1-\gamma_2 &= 0 \\ \tfrac{1}{2}-\gamma_2\alpha &= 0 \\ \tfrac{1}{2}-\gamma_2\beta &= 0 \end{aligned}$$

Note that we cannot choose $\gamma_2 = 0$, as that would lead to a contradiction in the last two equations. Since there are 3 equations and 4 variables, we let γ_2 be unspecified and then solve for α, β, γ_1 in terms of γ_2. This yields

$$\alpha = \beta = \frac{1}{2\gamma_2}, \quad \gamma_1 = 1-\gamma_2$$

Case: $\gamma_2 = \frac{1}{2}$

$$\begin{aligned} y_{n+1} = y_n + \tfrac{h}{2}\,[&f(x_n, y_n) \\ &+f(x_n+h, y_n+hf(x_n,y_n))] \end{aligned}$$

This is one iteration of the trapezoidal rule with an Euler predictor.

Case: $\gamma_2 = 1$.

$$y_{n+1} = y_n + hf(x_n+\tfrac{h}{2}, y_n+\tfrac{h}{2}f(x_n,y_n))$$

We can derive other second order formulas by choosing other values for γ_2. Sometimes this is done by attempting to minimize the next term in the truncation error. In the case of the above formula,

$$T_n(y) = c_3 h^3 + \cdots$$

with

$$\begin{aligned} c_3 &= \left(\tfrac{1}{6} - \tfrac{1}{2}\gamma_2\alpha^2\right) f_{xx} + \left(\tfrac{1}{3} - \gamma_2\alpha\beta\right) f_{xy} f \\ &\quad + \left(\tfrac{1}{6} - \tfrac{1}{2}\gamma_2\beta^2\right) f_{yy} f^2 + \tfrac{1}{6} f_y f_x + \tfrac{1}{6} f_y^2 f \end{aligned}$$

We can regard this as the dot product of two vectors:

$$c_3 = \begin{bmatrix} \frac{1}{6} - \frac{1}{2}\gamma_2\alpha^2 \\ \frac{1}{3} - \gamma_2\alpha\beta \\ \frac{1}{6} - \frac{1}{2}\gamma_2\beta^2 \\ \frac{1}{6} \\ \frac{1}{6} \end{bmatrix} \cdot \begin{bmatrix} f_{xx} \\ f_{xy} f \\ f_{yy} f^2 \\ f_y f_x \\ f_y^2 f \end{bmatrix}$$

Then

$$|c_3| \leq d_1(\gamma_2)\, d_2(f)$$

with

$$d_1(\gamma_2) = \Big[\Big(\tfrac{1}{6} - \tfrac{1}{2}\gamma_2\alpha^2\Big)^2 + \Big(\tfrac{1}{3} - \gamma_2\alpha\beta\Big)^2 + \Big(\tfrac{1}{6} - \tfrac{1}{2}\gamma_2\beta^2\Big)^2 + \tfrac{1}{18}\Big]^{\frac{1}{2}}$$

and $d_2(f)$ the lenth of the second vector. Pick γ_2 to minimize $d_1(\gamma_2)$. This occurs at $\gamma_2 = \frac{3}{4}$, and this yields the formula

$$y_{n+1} = y_n + \tfrac{h}{4}\big[f(x_n, y_n) + 3f(x_n + \tfrac{2}{3}h, y_n + \tfrac{2}{3}hf(x_n, y_n))\big]$$

3-STAGE FORMULAS

We have just studied 2-stage formulas. To obtain a higher rate of convergence, we must use more derivative evaluations. A 3-stage formula looks like

$$y_{n+1} = y_n + hF(x_n, y_n, h; f)$$

$$F(x, y, h; f) = \gamma_1 V_1 + \gamma_2 V_2 + \gamma_3 V_3$$

$$\begin{aligned} V_1 &= f(x, y) \\ V_2 &= f(x + \alpha_2 h, y + \beta_{2,1} h V_1) \\ V_3 &= f(x + \alpha_3 h, y + \beta_{3,1} h V_1 + \beta_{3,2} h V_2) \end{aligned}$$

Again it can be analyzed by expanding the truncation error

$$T_n(Y) = Y(x + h) - [Y(x) + hF(x, Y(x), h; f)]$$

in powers of h. Then the coefficients are determined by setting the lead coefficients to zero. This can be generalized to dealing with p-stage formulas. The algebra becomes very complicated; and the equations one obtains turn out to be dependent.

p-STAGE FORMULAS

We have just studied 2-stage formulas. To obtain a higher rate of convergence, we must use more derivative evaluations. A p-stage formula looks like

$$y_{n+1} = y_n + hF(x_n, y_n, h; f)$$

$$F(x, y, h; f) = \sum_{j=1}^{p} \gamma_i V_i$$

$$\begin{aligned} V_1 &= f(x, y) \\ V_2 &= f(x + \alpha_2 h, y + \beta_{2,1} h V_1) \\ &\vdots \\ V_p &= f\left(x + \alpha_p h, y + h \sum_{j=1}^{p-1} \beta_{p,j} V_j\right) \end{aligned}$$

Again it can be analyzed by expanding the truncation error

$$T_n(Y) = Y(x + h) - [Y(x) + hF(x, Y(x), h; f)]$$

in powers of h. Then the coefficients are determined by setting the lead coefficients to zero.

How high an order can be obtained?

A number of years ago, John Butcher derived the results shown in the following table.

p	1	2	3	4	5	6	7	8
Max order	1	2	3	4	4	5	6	6

This is counter to what people had believed; and it has some important consequences. We will return to this when considering the definition of some new Runge-Kutta formulas.

A POPULAR FOURTH ORDER METHOD

A popular classical formula uses

$$y_{n+1} = y_n + hF(x_n, y_n, h; f)$$

$$F(x, y, h; f) = \frac{1}{6}\left[V_1 + 2V_2 + 2V_3 + V_4\right]$$

$$\begin{aligned} V_1 &= f(x, y) \\ V_2 &= f(x + \tfrac{1}{2}h, y + \tfrac{1}{2}hV_1) \\ V_3 &= f(x + \tfrac{1}{2}h, y + \tfrac{1}{2}hV_2) \\ V_4 &= f\left(x + h, y + hV_3\right) \end{aligned}$$

For $y' = f(x)$, this becomes

$$y_{n+1} = y_n + \frac{h}{6}\Big[f(x) + 4f(x + \tfrac{1}{2}h) + f(x + h)\Big]$$

which is simply Simpson's integration rule.

It can be shown that the truncation error is

$$T_n(Y) = c(x_n)h^5 + O(h^6)$$

for a suitable function $c(x)$. In addition, we can show

$$|Y(x_n) - y_h(x_n)| \leq d(x_n)h^4$$

with a suitable $d(x)$. Finally, one can prove an asymptotic error formula

$$Y(x) - y_h(x) = D(x)h^4 + O(h^5)$$

This leads to the Richardson error estimate

$$Y(x) - y_h(x) \approx \frac{1}{15}\left[y_h(x) - y_{2h}(x)\right]$$

EXAMPLE. Solve

$$y' = \frac{1}{1+x^2} - 2y^2, \quad y(0) = 0$$

Its true solution is

$$Y(x) = \frac{x}{1+x^2}$$

Use stepsizes $h = .25$ and $2h = .5$.

CONVERGENCE

For the numerical method

$$y_{n+1} = y_n + hF(x_n, y_n, h; f)$$

we define the truncation error as

$$T_n(Y) = Y(x_{n+1}) - [Y(x_n) + hF(x_n, Y(x_n), h; f)]$$

with $Y(x)$ the true solution of

$$y' = f(x, y), \quad y(x_0) = Y_0$$

Introduce

$$\begin{aligned} \tau_n(Y) &= \frac{1}{h} T_n(Y) \\ &= \frac{Y(x_{n+1}) - Y(x_n)}{h} - F(x_n, Y(x_n), h; f) \end{aligned}$$

We will want $\tau_n(Y) \to 0$ as $h \to 0$. In this connection, introduce

$$\delta(h) = \max_{\substack{x_0 \le x \le b \\ -\infty < y < \infty}} |f(x, y) - F(x, y, h; f)|$$

and assume

$$\delta(h) \to 0 \quad \text{as} \quad h \to 0$$

Note then that

$$\begin{aligned} \tau_n(Y) &= \left[\frac{Y(x_{n+1}) - Y(x_n)}{h} - Y'(x_n)\right] \\ &\quad + [f(x_n, Y(x_n)) - F(x_n, Y(x_n), h; f)] \end{aligned}$$

The first term on the right side goes to zero from the definition of derivative; and the second term is bounded by $\delta(h)$, and thus it too goes to zero with h.

Returning to

$$T_n(Y) = Y(x_{n+1}) - [Y(x_n) + hF(x_n, Y(x_n), h; f)]$$

we rewrite it as

$$Y(x_{n+1}) = Y(x_n) + hF(x_n, Y(x_n), h; f) + h\tau_n(Y)$$

The numerical method is

$$y_{n+1} = y_n + hF(x_n, y_n, h; f)$$

To analyze the convergence, we subtract to get

$$\begin{aligned} e_{n+1} &= e_n + h\left[F(x_n, Y_n, h; f) - F(x_n, y_n, h; f)\right] \\ &\quad + h\tau_n(Y) \end{aligned}$$

To continue with this approach, we need to know what happens to F when the y argument is perturbed. In particular, we assume

$$|F(x, y, h; f) - F(x, z, h; f)| \leq L\,|y - z|$$

for all $x_0 \leq x \leq b$, $-\infty < y, z < \infty$, and all small values of h, say $0 < h \leq h_0$ for some h_0. This generally can be derived from the Lipschitz condition for the function $f(x, y)$.

For example, recall the second order method

$$\begin{aligned} y_{n+1} = y_n + \tfrac{h}{2}\,&[f(x_n, y_n) \\ &+ f(x_n + h, y_n + hf(x_n, y_n))] \end{aligned}$$

in which

$$F(x, y, h; f) = \frac{1}{2}f(x, y) + \frac{1}{2}f(x + h, y + hf(x, y))$$

Then

$$\begin{aligned} F(x,y,h;f) - F(x,z,h;f) \\ = \tfrac{1}{2}\left[f(x,y) - f(x,z)\right] \\ + \tfrac{1}{2}\left[f(x+h, y+hf(x,y)) \right. \\ \left. - f(x+h, z+hf(x,z))\right] \end{aligned}$$

Use the Lipschitz condition on f:

$$\begin{aligned} |F(x,y,h;f) - F(x,z,h;f)| \\ \leq \tfrac{1}{2}\left|f(x,y) - f(x,z)\right| \\ + \tfrac{1}{2}\left|f(x+h, y+hf(x,y)) \right. \\ \left. - f(x+h, z+hf(x,z))\right| \\ \leq \tfrac{1}{2}K\left|y-z\right| + \tfrac{1}{2}K\left[|y-z| \right. \\ \left. + h\left|f(x,y) - f(x,z)\right|\right] \end{aligned}$$

This leads to

$$|F(x,y,h;f) - F(x,z,h;f)| \leq \left[K + K^2\frac{h}{4}\right]|y-z|$$

For $h \leq 1$, take $L = K\,(1 + K/4)$. Then

$$|F(x,y,h;f) - F(x,z,h;f)| \leq L\,|y-z|$$

Return to the error equation

$$\begin{aligned} e_{n+1} &= e_n + h\left[F(x_n, Y_n, h; f) - F(x_n, y_n, h; f)\right] \\ &\quad + h\tau_n(Y) \end{aligned}$$

Taking bounds,

$$|e_{n+1}| \leq |e_n| + hL\,|e_n| + h\tau(h)$$

with

$$\tau(h) = \max_{x_0 \leq x_n \leq b} |\tau_n(Y)|$$

We analyze

$$|e_{n+1}| \leq (1 + hL)\,|e_n| + h\tau(h)$$

exactly as was done for Euler's method. This yields

$$|Y(x_n) - y_n| \leq e^{(b-x_0)L}\,|e_0| + \frac{e^{(b-x_0)L} - 1}{L}\tau(h)$$

Generally, $e_0 = 0$; and the speed of convergence is that of $\tau(h)$.

STABILITY

Using the ideas introduced above, we can do a stability and rounding error analysis that is essentially the same as that done for Euler's method. We omit it here.

Consider now applying a Runge-Kutta method to the model equation

$$y' = \lambda y, \quad y(0) = 1$$

As an example, use the second order method

$$y_{n+1} = y_n + hf(x_n + \tfrac{h}{2}, y_n + \tfrac{h}{2} f(x_n, y_n))$$

This yields

$$\begin{aligned} y_{n+1} &= y_n + h\lambda \left[y_n + \tfrac{h}{2} f(x_n, y_n) \right] \\ &= y_n + h\lambda \left[y_n + \tfrac{h}{2} \lambda y_n \right] \\ &= y_n + h\lambda y_n + \tfrac{1}{2} (h\lambda)^2 y_n \\ &= \left[1 + h\lambda + \tfrac{1}{2} (h\lambda)^2 \right] y_n \end{aligned}$$

Thus

$$y_n = \left[1 + h\lambda + \tfrac{1}{2} (h\lambda)^2 \right]^n, \quad n \geq 0$$

If we consider $\lambda < 0$ or real$(\lambda) < 0$, we want to know when $y_n \longrightarrow 0$. That will be true if

$$\left|1 + h\lambda + \tfrac{1}{2}(h\lambda)^2\right| < 1$$

For λ real, this is true if

$$-2 < h\lambda < 0$$

This is the region of absolute stability for this second order Runge-Kutta method. It is not much different than that of some of the Adams family of methods.

It can be shown that there are no A-stable explicit Runge-Kutta methods. Thus explicit Runge-Kutta methods are not suitable for stiff differential equations.

ERROR CONTROL

Again we want to estimate the local error in solving the differential equation. In particular, let $u_n(x)$ denote the true solution of the problem

$$y' = f(x, y), \quad x \geq x_n, \quad y(x_n) = y_n$$

In the text, on page 428, we derive the formula

$$\begin{aligned} & u_n(x_n + 2h) - y_h(x_n + 2h) \\ & \quad \doteq \frac{1}{2^m - 1} \left[y_h(x_n + 2h) - y_{2h}(x_n + 2h) \right] \end{aligned}$$

where m is the order of the Runge-Kutta method. We denote the error estimate on the right by *trunc*. Then for error control purposes, as earlier with *Detrap*, we want to have *trunc* satisfy a local error control per unit stepsize:

$$.5\epsilon h \leq |trunc| \leq 2\epsilon h$$

If this violated, then we seek a new stepsize $\widehat{h}$ to have $trunc = \epsilon\widehat{h}$, and we continue with the solution process.

We could also use the extrapolated solution

$$\widetilde{y}_h(x_n + 2h) = y_h(x_n + 2h) + trunc$$

This would have to be analyzed for stability and convergence. But with it, an error per step criteria applied to $y_h(x_n + 2h)$,

$$.5\epsilon \leq |trunc| \leq 2\epsilon$$

while keeping $\widetilde{y}_h(x_n + 2h)$ will usually suffice. Many codes do something of this kind.

COST IN FUNCTION EVALUATIONS

We are advancing from x_n to x_{n+2}. what does this cost, if we include calculation of *trunc*? Consider it for the classical RK method

$$y_{n+1} = y_n + hF(x_n, y_n, h; f)$$

$$F(x, y, h; f) = \frac{1}{6}\left[V_1 + 2V_2 + 2V_3 + V_4\right]$$

$$\begin{aligned} V_1 &= f(x, y) \\ V_2 &= f(x + \tfrac{1}{2}h, y + \tfrac{1}{2}hV_1) \\ V_3 &= f(x + \tfrac{1}{2}h, y + \tfrac{1}{2}hV_2) \\ V_4 &= f\left(x + h, y + hV_3\right) \end{aligned}$$

Calculating $y_h(x_n + h)$: 4 new evaluations of f
Calculating $y_h(x_n + 2h)$: 4 new evaluations of f
Calculating $y_{2h}(x_n + 2h)$: 3 new evaluations of f
Total evaluations: 11

For comparison, an implicit multistep method will typically use 4 evaluations to go from x_n to x_{n+2}.

RUNGE-KUTTA-FEHLBERG METHODS

There are 6-stage methods of order 5. Fehlberg thought of imbedding a order 4 method into the 6-stage method, and to then use of the difference of the two solutions to estimate the local error in lower order method. In particular, he introduced methods

$$\begin{aligned} y_{n+1} &= y_n + h\sum_{j=1}^{5}\gamma_j V_j \\ \widehat{y}_{n+1} &= y_n + h\sum_{j=1}^{6}\widehat{\gamma}_j V_j \end{aligned}$$

$$\begin{aligned} V_1 &= f(x_n, y_n) \\ V_i &= f\left(x + \alpha_i h, y + h\sum_{j=1}^{i-1}\beta_{i,j}V_j\right),\ i = 2, ..., 6 \end{aligned}$$

He showed the coefficients could be chosen so that y_{n+1} was an order 4 method and $\widehat{y}_{n+1}$ was an order 5 method. Then he estimated the local error using

$$u_n(x_{n+1}) - y_{n+1} \approx \widehat{y}_{n+1} - y_{n+1}$$

DISCUSSION

In determining the formulas for y_{n+1} and $\widehat{y}_{n+1}$, Fehlberg had some additional freedom in choosing the coefficients. He looked at the leading terms in the expansion of the errors

$$u_n(x_{n+1}) - y_{n+1}, \quad u_n(x_{n+1}) - \widehat{y}_{n+1}$$

in powers of h, say

$$u_n(x_{n+1}) - y_{n+1} = c_5 h^5 + c_6 h^6 + \cdots$$

$$u_n(x_{n+1}) - \widehat{y}_{n+1} = \widehat{c}_6 h^6 + \widehat{c}_7 h^7 + \cdots$$

He chose the coefficients so as
(1) to make $\widehat{y}_{n+1} - y_{n+1}$ an accurate estimator of $u_n(x_{n+1}) - y_{n+1}$,

$$\frac{[u_n(x_{n+1}) - y_{n+1}] - [\widehat{y}_{n+1} - y_{n+1}]}{u_n(x_{n+1}) - y_{n+1}} = \frac{\widehat{c}_6 h^6 + \widehat{c}_7 h^7 + \cdots}{c_5 h^5 + c_6 h^6 + \cdots} \approx \frac{\widehat{c}_6}{c_5} h$$

(2) to make c_5 as small as possible, while having

$$c_5 \geq c_6$$

His choice of coefficients for the order (4,5) pair of formulas is given in the text.

EXAMPLE. Apply the fourth order formula to the simple problem

$$y' = x^4, \quad y(0) = 0$$

The RKF method of order 4 has the truncation error

$$T_n(Y) \doteq 0.00048h^5$$

The classical RK method, discussed earlier, has the truncation error

$$T_n(Y) \doteq -0.0083h^5$$

Thus the RKF method appears to be more accurate than the classical formula.

COSTS For its computational cost when stepping from x_n to x_{n+2}, 12 evaluations of f are required, as compared to the 11 evaluations required in estimating the local error in the classical fourth order method. In fact, we do not need to take two steps to estimate the local error, but can in fact change stepsize at each x_n if so needed.

PROGRAMS

In *MATLAB*, look at the programs *ode23* and *ode45*. Also look at the associated demo programs.

In the class account, look at the Fortran program *fehlberg-45.f*. It takes a single step, returning both y_{n+1} and the estimated local error $\widehat{y}_{n+1} - y_{n+1}$.

In addition, I have included the program *rkf45.f*, which implements the Fehlberg (4,5) pair with automatic control of the local error. It is from Sandia National Labs, and it is a very popular implementation.

IMPLICIT RUNGE-KUTTA METHODS

As before, define a p-stage method

$$y_{n+1} = y_n + hF(x_n, y_n, h; f)$$

$$F(x, y, h; f) = \sum_{j=1}^{p} \gamma_i V_i$$

But now, let

$$V_i = f\left(x + \alpha_i h, y + h\sum_{j=1}^{p} \beta_{i,j} V_j\right), \quad i = 1, ..., p$$

Thus we must solve p equations in p unknowns at each step of solving the differential equation.

The reason for doing this is that it leads to numerical methods with better stability properties when solving stiff differential equations. In particular, we can obtain A-stable methods of any desired order.

COLLOCATION METHODS

An important category of implicit Runge-Kutta methods are obtained as follows. Let $p \geq 1$ be an integer, and let the points

$$0 \leq \alpha_1 < \alpha_2 < \cdots < \alpha_p \leq 1$$

be given. Assuming we are at the point (x_n, y_n), find a polynomial $q(x)$ of degree p for which

$$q(x_n) = y_n$$

$$q'(x_n + \alpha_i h) = f\left(x_n + \alpha_i h, q(x_n + \alpha_i h)\right)$$

for $i = 1, ..., p$. Then this is equivalent to an implicit Runge-Kutta method; and it is also called a *block by block method*.

ORDER OF CONVERGENCE

Assume $\{\alpha_i\}$ have been so chosen that

$$\int_0^1 \omega(\tau)\tau^j \, d\tau = 0, \quad j = 0, 1, ..., m-1$$

$$\omega(\tau) = (\tau - \alpha_1) \cdots (\tau - \alpha_p)$$

Then for our collocation method,

$$T_n(Y) = O(h^{p+m+1})$$

and the order of convergence is $p + m$.

Defining $\{\alpha_1, ..., \alpha_p\}$ as the zeros of the Legendre polynomial of degree p, normalized to $[0, 1]$, leads to a method of order $2p$. These methods are A-stable for every $p \geq 1$.

ADDITIONAL REFERENCES

Larry Shampine, *Numerical Solution of Ordinary Differential Equations*, Chapman & Hall Publishers, 1994.

Larry Shampine and Mark Reichelt, "The Matlab ODE Suite", *SIAM Journal On Scientific Computing* 18 (1997), pp. 1-22.

Arieh Iserles, *Numerical Analysis of Differential Equations*, Cambridge University Press, 1996. This also includes as extensive development on the numerical solution of partial differential equations.

Uri Ascher, Robert Mattheij, and Robert Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, Prentice-Hall, 1988.

Uri Ascher and Linda Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM Pub., 1998.